


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

Portabl + "password attempts"+ GUI

SEARCH

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **Portabl password attempts GUI**

Found 64 of 185,942

Sort results by

 relevance ☒

Display results

 expanded form ☒
[Save results to a Binder](#)
[Search Tips](#)
☐ Open results in a new window

 Try an [Advanced Search](#)  
 Try this search in [The ACM Guide](#)

Results 1 - 20 of 64

 Result page: [1](#) [2](#) [3](#) [4](#) [next](#)

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Session 1: engineering applications I: A scalable high-performance environment for](#)

#### [fluid flow analysis on unstructured grids](#)

Deb Banerjee, Thomas Tysinger, Wayne Smith

 November 1994 **Proceedings of the 1994 ACM/IEEE conference on Supercomputing**

Publisher: ACM Press

 Full text available: pdf(1.38 MB) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper describes an integrated environment for the analysis and solution of fluid flow problems, and contains a Computational Fluid Dynamics (CFD) solver and visualization system as its major components. The flow solver is capable of solving the Navier-Stokes equations about complex geometries through the use of unstructured, solution adaptable, grids. Unlike post-processing visualization environments, the visualization system is designed to work in a co-processing mode with the CFD applicat ...

### 2 [Authentication using graphical passwords: effects of tolerance and image choice](#)

#### Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, Nasir Memon

 July 2005 **Proceedings of the 2005 symposium on Usable privacy and security SOUPS '05**

Publisher: ACM Press

 Full text available: pdf(555.83 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Graphical passwords are an alternative to alphanumeric passwords in which users click on images to authenticate themselves rather than type alphanumeric strings. We have developed one such system, called PassPoints, and evaluated it with human users. The results of the evaluation were promising with respect to rmemorability of the graphical password. In this study we expand our human factors testing by studying two issues: the effect of tolerance, or margin of error, in clicking on the password ...

**Keywords:** PassPoints, authentication, graphical passwords, human factors, password images, password security, tolerance, usable security

### 3 [Doctoral symposium: posters: Developing cost-effective model-based techniques for](#)

#### [GUI testing](#)

Qing Xie


 May 2006 **Proceeding of the 28th international conference on Software engineering**
**ICSE '06**

Publisher: ACM Press

Full text available:  [pdf\(116.69 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Most of today's software users interact with the software through a graphical user interface (GUI). While GUIs have become ubiquitous, testing of GUIs has remained until recently, a neglected research area. Existing GUI testing techniques are extremely resource intensive primarily because GUIs have very large input spaces. This research proposes to advance the state-of-the-art in GUI testing by empirically studying GUI faults, interactions between GUI events, why certain event interactions lead ...

#### 4 Coverage criteria for GUI testing

 Atif M. Memon, Mary Lou Soffa, Martha E. Pollack  
September 2001 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-9**, Volume 26 Issue 5

**Publisher:** ACM Press

Full text available:  [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A widespread recognition of the usefulness of graphical user interfaces (GUIs) has established their importance as critical components of today's software. GUIs have characteristics different from traditional software, and conventional testing techniques do not directly apply to GUIs. This paper's focus is on coverage criteria for GUIs, important rules that provide an objective measure of test quality. We present new coverage criteria to help determine whether a GUI has been adequately tested. ...

**Keywords:** GUI test coverage, GUI testing, component testing, event-based coverage, event-flow graph, integration tree

#### 5 Java Virtual Machine: An examination of the run-time performance of GUI creation frameworks

Christopher J. Howell, Gregory M. Kapfhammer, Robert S. Roos  
June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java PPPJ '03**

**Publisher:** Computer Science Press, Inc.


Full text available:  [pdf\(108.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

The graphical user interface (GUI) is an important component of many software systems. Past surveys indicate that the development of a GUI is a significant undertaking and that the GUI's source code often comprises a substantial portion of the program's overall source base. Graphical user interface creation frameworks for popular object-oriented programming languages enable the rapid construction of simple and complex GUIs. In this paper, we examine the run-time performance of two GUI creation f ...

#### 6 Do assignments with required GUI's help students learn?

Terry Scott  
October 2003 **Journal of Computing Sciences in Colleges**, Volume 19 Issue 1

**Publisher:** Consortium for Computing Sciences in Colleges

Full text available:  [pdf\(33.90 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Student learning in a class depends on time on task. Many students study harder on assignments that interest them. It was postulated that programming assignment with GUI's required would interest them more, leading to more time on task and consequently more learning. This paper describes a study done on a class to determine student attitudes about programming assignments requiring a GUI.

7 Technical papers: design recovery: Browsing and searching source code of applications written using a GUI framework



Amir Michail

May 2002 **Proceedings of the 24th International Conference on Software Engineering**

**Publisher:** ACM Press

Full text available: pdf(1.71 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Nowadays, applications are typically written using an object-oriented GUI framework. In this paper we explore the possibility of using the GUI of such applications to guide browsing and search of their source code. Such a tool would be helpful for software maintenance and reuse, particularly when the application source is unfamiliar. Intuitively, we would expect the task of browsing and searching source code of an application written using a GUI framework to be easier than one that doesn't because ...

8 Generating the user interface: GUI generation from annotated source code



Josef Jelinek, Pavel Slavik

November 2004 **Proceedings of the 3rd annual conference on Task models and diagrams TAMODIA '04**

**Publisher:** ACM Press

Full text available: pdf(153.75 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Creating user interfaces is a common task in application development. It can become time and money consuming if the same application is to be run on more platforms with different restrictions and requirements. To reduce the development cost and time the user interface can be defined on an abstract level in the form of a task model. Explicit defining and maintaining the task model can complicate the development especially in its early stages when application prototype are built. We present a way ...

**Keywords:** automatic GUI generation, visual programming

9 Is GUI programming a database research problem?



Nita Goyal, Charles Hoch, Ravi Krishnamurthy, Brian Meckler, Michael Suckow

June 1996 **ACM SIGMOD Record , Proceedings of the 1996 ACM SIGMOD international conference on Management of data SIGMOD '96**, Volume 25 Issue 2

**Publisher:** ACM Press

Full text available: pdf(1.48 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Programming nontrivial GUI applications is currently an arduous task. Just as the use of a declarative language simplified the programming of database applications, we ask whether we can do the same for GUI programming? Can we then import a large body of knowledge from database research? We answer these questions by describing our experience in building nontrivial GUI applications initially using C++ programming and subsequently using Logic++, a higher order Horn clause logic language on complex ...

10 Software engineering #2: Specification-driven automated testing of GUI-based Java programs



Yanhong Sun, Edward L. Jones

April 2004 **Proceedings of the 42nd annual Southeast regional conference**

**Publisher:** ACM Press

Full text available: pdf(503.31 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents a specification-driven approach to test automation for GUI-based JAVA programs as an alternative to the use of capture/replay. The NetBeans Jemmy

library provides the basic technology. We introduce a GUI-event test specification language from which an automated test engine is generated. The test engine uses the library and incorporates the generation of GUI events, the capture of event responses, and an oracle to verify successful completion of events. The engine, once genera ...

**Keywords:** test automation, test engine, test specification language

#### 11 An overview of portable GUI software



Wade Guthrie

January 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 1

**Publisher:** ACM Press

Full text available: [pdf\(1.90 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

This article attempts to bring together as much information as possible concerning platform-independent Graphical User Interface (PIGUI) development kits. It is based on a FAQ list (answers to Frequently Answered Questions) maintained and periodically updated as a service to the net by the author. What is presented here is a number of tables summarizing available PIGUI's, followed by descriptions of the individual products, with reviews and users' comments where possible.

#### 12 Window real objects: a distributed shared memory for distributed implementation of GUI applications



Noboru Koshizuka, Ken Sakamura

December 1993 **Proceedings of the 6th annual ACM symposium on User interface software and technology**

**Publisher:** ACM Press

Full text available: [pdf\(1.31 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** BTRON, distributed shared memory, graphical user interface, multiuser interface, parallel programming, window system

#### 13 Developing principles of GUI programming using views



Judith Bishop, Nigel Horspool

March 2004 **ACM SIGCSE Bulletin , Proceedings of the 35th SIGCSE technical symposium on Computer science education SIGCSE '04**, Volume 36 Issue 1

**Publisher:** ACM Press

Full text available: [pdf\(262.89 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper proposes that GUI development is as important as other aspects of programming, such as a sound understanding of control structures and object orientation. Far less attention has been paid to the programming structures for GUIs and certainly there are few cross language principles to aid the programmer. We propose that principles of GUIs can be extracted and learnt, and that they do enhance good programming practice. These principles have been implemented in our Views system which feat ...


**Keywords:** XML, event-based programming, graphical user interfaces, platform independence

#### 14 The Importance of the GUI in Cross Platform Development

Michael Babcock

May 1998 **Linux Journal**

**Publisher:** Specialized Systems Consultants, Inc.

Full text available:  [html\(23.51 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The fragmentation of development energy into too many GUI toolkits is one of the most serious problems facing the Linux community today

# 15 A visual test development environment for GUI systems



Thomas Ostrand, Aaron Anodide, Herbert Foster, Tarak Goradia

March 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1998 ACM SIGSOFT international symposium on Software testing and analysis**

**ISSTA '98**, Volume 23 Issue 2

**Publisher:** ACM Press

Full text available:  [pdf\(2.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We have implemented an experimental test development environment (TDE) intended to raise the effectiveness of tests produced for GUI systems, and raise the productivity of the GUI system tester. The environment links a test designer, a test design library, and a test generation engine with a standard commercial capture/replay tool. These components provide a human tester the capabilities to capture sequences of interactions with the system under test (SUT), to visually manipulate and modify the s ...

**Keywords:** GUI-based system, capture/replay, test coverage, test designer, test generation, test maintenance, test scenario, testing, visual editor

# 16 Participatory GUI design from task models



Tom Dayton, Joseph Kramer, Al McFarland, Monica Heidelberg

April 1996 **Conference companion on Human factors in computing systems: common ground**

**Publisher:** ACM Press

Full text available:  [pdf\(229.81 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

**Keywords:** HCI, PANDA, PD, UCD, human-computer interaction, prototyping, usability engineering, user-centered design

# 17 TelePICTIVE: computer-supported collaborative GUI design for designers with diverse expertise



David S. Miller, John G. Smith, Michael J. Muller

December 1992 **Proceedings of the 5th annual ACM symposium on User interface software and technology**

**Publisher:** ACM Press

Full text available:  [pdf\(1.04 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

It is generally accepted that it is important to involve the end users of a Graphical User Interface (GUI) in all stages of its design and development. However, traditional GUI development tools typically do not support collaborative design. TelePICTIVE is an experimental software prototype designed to allow computer-naive users to collaborate with experts at possibly remote locations in designing GUIs. TelePICTIVE is based on the PICTIVE participatory design methodology, and has ...

**Keywords:** CSCW, MUMMS application, PICTIVE, collaborative, graphical user interface, multi-user, participatory design

18 Session II: wxHaskell: a portable and concise GUI library for haskell ☐

Daan Leijen

September 2004 **Proceedings of the 2004 ACM SIGPLAN workshop on Haskell Haskell '04****Publisher:** ACM PressFull text available: pdf(179.83 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

wxHaskell is a graphical user interface (GUI) library for Haskell that is built on wxWidgets: a free industrial strength GUI library for C++ that has been ported to all major platforms, including Windows, Gtk, and MacOS X. In contrast with many other libraries, wxWidgets retains the native look-and-feel of each particular platform. We show how distinctive features of Haskell, like parametric polymorphism, higher-order functions, and first-class computations, can be used to present a concise and ...

**Keywords:** C++, Haskell, combinator library, graphical user interface, layout, wxWidgets

19 Bridge for buttons - a GUI design methodology applied in non-GUI consumer product design ☐

Simo Säde, Katja Battarbee

August 2000 **Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques****Publisher:** ACM PressFull text available: pdf(363.24 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper describes the experience gained in the case study of adapting the Bridge, a GUI design method to the design of non-GUI interactive consumer products. An industrial design consultancy was looking for an efficient way of discussing the initial design ideas for a product and its user interface (UI) with the client and the client's software design consultant. The Bridge, which is a fast participatory method for bridging the gap between user requirements and the design of ob ...

**Keywords:** case study, discount usability, industrial and user interface design techniques, the bridge

20 Qt GUI Toolkit: Porting graphics to multiple platforms using a GUI toolkit ☐

Eirik Eng

November 1996 **Linux Journal****Publisher:** Specialized Systems Consultants, Inc.Full text available: html(36.64 KB) Additional Information: [full citation](#), [index terms](#)

Results 1 - 20 of 64

Result page: [1](#) [2](#) [3](#) [4](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)

## WEST Search History

DATE: Friday, September 29, 2006

### Hit Count

#### Hide? Set Name Query

*DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR*

<input type="checkbox"/>	L49	(portable and (password near3 valid\$2)and (user near10 (graphical near2 user))).clm.	1
<input type="checkbox"/>	L48	(portable and (password near3 valid\$2)and (user near10 GUI)).clm.	0
<input type="checkbox"/>	L47	(portable and (password near3 attempt\$2)and (user near10 GUI)).clm.	0
<input type="checkbox"/>	L46	(L42 or L43 or L44) and ((eras\$4 or delet\$6) near6 memory same password)	52
<input type="checkbox"/>	L45	(L42 or L43 or L44) and (portable same (eras\$4 or delet\$6) same password same tamper)	0
<input type="checkbox"/>	L44	(726/26).ccls.	1017
<input type="checkbox"/>	L43	(380/281).ccls.	249
<input type="checkbox"/>	L42	(380/277  713/193  713/194  713/182  713/183).ccls.	4069
<input type="checkbox"/>	L41	4849927 and portable	4
<input type="checkbox"/>	L40	4849927 and poratable	0
<input type="checkbox"/>	L39	4849927	37
<input type="checkbox"/>	L38	4849927 and password	2
<input type="checkbox"/>	L37	L36 same encrypt\$8	1
<input type="checkbox"/>	L36	(eras\$7 near3 memory near10 invalid)	260
<input type="checkbox"/>	L35	L34 and portable	7
<input type="checkbox"/>	L34	arani and eras\$7	21
<input type="checkbox"/>	L33	eras\$6 near3 encrypted adj content and overwrit\$8	3
<input type="checkbox"/>	L32	eras\$6 near3 encrypted adj content	14
<input type="checkbox"/>	L31	eras\$6 near3 encrypted adj content and valid near3 password	0
<input type="checkbox"/>	L30	eras\$6 near3 encrypted adj content same password	0
<input type="checkbox"/>	L29	eras\$6 near3 secure adj content same password	0
<input type="checkbox"/>	L28	portable and (valid near4 password same eras\$7)	18
<input type="checkbox"/>	L27	portable and (valid adj password same eras\$7)	5
<input type="checkbox"/>	L26	portable same valid adj password same eras\$7	3
<input type="checkbox"/>	L25	portable same password same eras\$7 same transmit\$8	8
<input type="checkbox"/>	L24	PDA same password same eras\$7	7
<input type="checkbox"/>	L23	PDA same password same ears\$7	0
<input type="checkbox"/>	L22	portable same password same (eras\$7 near3 data)	15
<input type="checkbox"/>	L21	portable same passowrd same eras\$7	0
<input type="checkbox"/>	L20	poratble same passowrd same eras\$7	0
<input type="checkbox"/>	L19	((eras\$7 or delet\$6) near3 application) and (password\$2 same portable))	20

## Search History Transcript

	<i>DB=PGPB,USPT; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L18 L17 and (eras\$7 or delet\$7)	5
	<i>DB=USPT,PGPB; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L17 ('4860005'  '5646604'  '5719523'  '5862472'  '6064627')![pn]	5
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L16 (((eras\$7 or delet\$6) near3 data) same password\$2 same portable)	18
<input type="checkbox"/>	L15 ((eras\$7 or delet\$6) same password\$2 same portable)	104
<input type="checkbox"/>	L14 L7 and ((eras\$7 or delet\$6) same password\$2)	89
<input type="checkbox"/>	L13 L7 and ((eras\$7 or delet\$6))	489
<input type="checkbox"/>	L12 L7 and (lock same password same (eras\$7 or delet\$6))	3
<input type="checkbox"/>	L11 L7 and (lock same password)	48
<input type="checkbox"/>	L10 L7 and lock	195
<input type="checkbox"/>	L9 L7 and ((eras\$6 or delet\$6 or destrot\$3) near5 application) and lock	3
<input type="checkbox"/>	L8 L7 and ((eras\$6 or delet\$6 or destrot\$3) near5 application)	25
<input type="checkbox"/>	L7 (726/16  713/182).ccls.	1194
<input type="checkbox"/>	L6 L4 and exceed\$6	15
<input type="checkbox"/>	L5 L4 same exceed\$6	0
<input type="checkbox"/>	L4 (reformat\$4 or (re adj format\$3)) same (storage or memory) same password	32
<input type="checkbox"/>	L3 L2 and (eras\$5 or delet\$5)	5
	<i>DB=USPT,PGPB; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L2 ('4860005'  '5646604'  '5719523'  '5862472'  '6064627')![pn]	5
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L1 portable same (password\$3 near3 exceed\$5)	3

END OF SEARCH HISTORY